Detection and Measurement of Syntactic Templates in Generated Text

Chantal Shaib[1] **Yanai Elazar**[2,3] **Junyi Jessy Li**[4] **Byron C. Wallace**[1]
[1]Northeastern University, [2]Allen Institute for AI,
[3]University of Washington, [4]The University of Texas at Austin
{shaib.c, b.wallace}@northeastern.edu

Abstract

The diversity of text can be measured beyond word-level features, however existing diversity evaluation focuses primarily on word-level features. Here we propose a method for evaluating diversity over syntactic features to characterize general repetition in models, beyond frequent n-grams. Specifically, we define *syntactic templates* (e.g., strings comprising parts-of-speech) and show that models tend to produce templated text in downstream tasks at a higher rate than what is found in human-reference texts We find that most (76%) templates in model-generated text can be found in pre-training data (compared to only 35% of human-authored text), and are not overwritten during fine-tuning or alignment processes such as RLHF. The connection between templates in generated text and the pre-training data allows us to analyze syntactic templates in models where we do not have the pre-training data. We also find that templates as features are able to differentiate between models, tasks, and domains, and are useful for qualitatively evaluating common model constructions. Finally, we demonstrate the use of templates as a useful tool for analyzing style memorization of training data in LLMs [1].

1 Introduction

An open question about large language models (LLMs) is what patterns such models learn from pre-training data (Goldberg, 2019; Petroni et al., 2019; Bender et al., 2021; Chen et al., 2024), and whether the same patterns appear generally across downstream tasks and datasets (Hupkes et al., 2023). While prior work has focused on the quality of generation (Zhang et al., 2019; Dou et al., 2022; Kryściński et al., 2020), and more recently on text generation novelty (McCoy et al., 2023; Merrill et al., 2024), there has been limited work on characterizing the sorts of lexical patterns that are learned by LLMs.

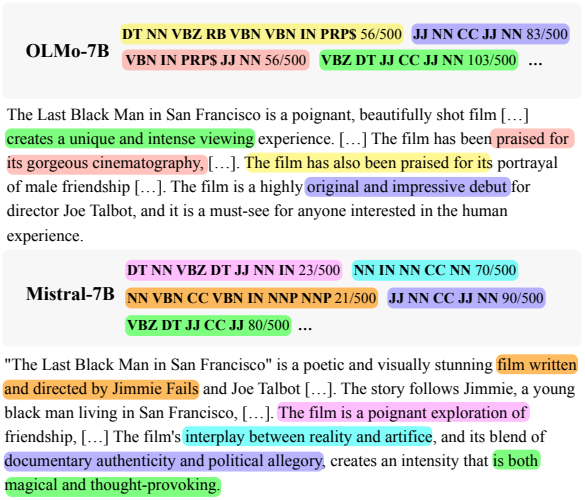

Figure 1: Sample movie meta-reviews generated by OLMo-7B (top) and Mistral-7B (bottom) by prompting the Rotten Tomatoes dataset. Templates appear at varying rates (frequency shown out of 500 generations), and differ across models. We extract templates from the entire corpus of generated text for each model, and match the text to the part-of-speech templates (highlighted), following by the frequency of each template.

Consider, for instance, the generated text from OLMo-Instruct in Figure 1, which is sampled from a corpus of movie review summaries. This was produced by prompting the model to summarize a collection of human-written movie reviews: "*The Last Black Man in San Francisco is a poignant, beautifully shot film [...] creates a unique and intense viewing experience [...]*". While this generated text was not seen in Dolma (Soldaini et al., 2024), OLMo's pre-training data, we find a total of 35 unique repeated sequences of part-of-speech (POS) tags of lengths $n = 5$ to 8 in the summarized movie reviews. Further, we find that 33 out of the 35 (95%) sequences appear in the pre-training data. As such, while the generated text itself is novel, it relies on common syntactic sequences seen in the training data.

In this work, we quantify and measure LLMs' us-

[1]https://cshaib.github.io/syntactic_templates/

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 6416–6431
November 12-16, 2024

ages of repetitive sequences in text generation. We introduce and focus on *syntactic templates*, namely POS sequences, a syntactic abstraction over texts. We seek to answer the following questions:

RQ1 To what extent do outputs generated by instruction-tuned LLMs contain templates?

RQ2 Can we locate model-generated templates in (pre-)training data?

RQ3 Can syntactic templates be used for detection of data memorization?

We start by introducing syntactic templates, and defining methods for detecting and measuring such templates in generated texts (§3). We evaluate eight models on three different tasks (§5). We show how training data templates are memorized and subsequently generated by models trained on them (§6). We then show how such insights allow one to draw conclusions about the training data used by closed models in a downstream summarization task (§7). Finally, we show that our metrics can also be used as a softer version of memorization. For instance, while Carlini et al. (2022) estimates that 1% of texts to be memorized, we find between 0.8 - 3.1% more *soft-memorized* texts over verbatim memorization, often by replacing numbers and synonyms (§8).

2 Related Work

Diversity in Text Generation Past evaluations of diversity in LLM outputs have primarily focused on token-level diversity (Montahaei et al., 2019; Bache et al., 2013). Diverse sampling strategies have been introduced to address the lower token diversity observed in neural text generation (Holtzman et al., 2020; Roberts et al., 2020), however it is unclear whether such sampling strategies also increase the diversity of the syntactic structure in LLMs. Beyond lexical diversity, Padmakumar and He (2023) extend definitions for measuring content diversity, which has broad applications in downstream tasks such as summarization and creative story generation. Recent work has quantified the drop in generated-text diversity specifically relative to the RLHF training process, however this again focused primarily on token level diversity (Kirk et al., 2024). Our work aligns more closely with the first body of work; we measure the syntactic structure of text rather than its semantic content. Most similar to our work is Bär et al. (2012), which broadly evaluates text repetition metrics at the stylistic, content, and lexical level. Our methods do not address repetition in content but rather focus on extending the characterization of lexical and stylistic repetition with text abstractions in LLMs.

Structural Analysis of Text DiMarco and Hirst (1993) provide a computational approach comprising lexical and syntactic components to describe stylistic elements in model-generated text. The discussion around style in writing has been adopted broadly for a variety of downstream tasks such as author or model attribution (Wu et al., 2023; Lample et al., 2018; Rosenfeld and Lazebnik, 2024). While our main goal is to provide measurements and characterizations of repetitive syntactic features in text, definitions of stylistic elements are closely related and help contextualize our findings. One can use our definitions of templates to ask broader questions about the prevailing syntactic style in a given corpus. Indeed, recent works adopt various measures of linguistic analysis to address differences in writing style in both human-written and model-generated texts (Krishna et al., 2020; Soler-Company and Wanner, 2017).

AI Text Detection In identifying n-gram features that appear in high frequencies in model-generated text, a natural question arises as to whether such features can be used to reliably *detect* model-generated text. Prior work has established that this is difficult, and that text-level features at the corpus level correlate with text being model-generated (Liang et al., 2024a,b). In this work, we make no claims for the use of templates in AI-text detection. Our aim is to *characterize* patterns rather than *detect* generated outputs, and to provide a basis for future work on model linguistic diversity.

3 Detecting Syntactic Templates

Our goal is to search for abstract representations of texts to capture more subtle repetitions than mere text memorization. Repeated strings of literal tokens may not be sufficient for describing such redundancy nor why a summary produced by, e.g., ChatGPT, might seem familiar.

Focusing on syntactic patterns rather than tokens allows us to capture such repetitions For example, a pattern consisting of the part-of-speech sequence `DT JJ NN IN DT JJ NN` will match to phrases in movie reviews (*"a romantic comedy about a corporate executive"*) and in news summarization

```
RB , DT NN VBD:  Fiercely , the beast breathed,
                 Slowly , the caterpillar inched,

VBD IN DT NN .:  creaked in the wind .,
                 emerged as a butterfly .,
                 inched along the branch .,
                 rattled with each gust .,

VBN IN JJ NNS .: diced into perfect cubes .,
                 groaned on rusty hinges .,
                 julienned into thin strips .,
                 minced into fine fragments .,
                 sliced into uniform rounds .
```

Figure 2: Example templates (left) and matched text (right) returned from the `diversity` package.

(*"a humorous insight into the perceived class"*), even though these sentences have only one token overlap.

3.1 Defining Templates

Given a sequence of tokens $T = (t_1, t_2, \ldots, t_n)$, and a function f that computes an abstraction over T (e.g., part-of-speech tags), we define a *template* as a sub-sequence of abstractions over the tokens $f(T)$ that repeats at least τ times in T. Figure 1 shows examples of templates and their counts across the Rotten Tomatoes dataset (Leone, 2020).

3.2 Extracting Syntactic Templates from Text

We operationalize the definition in 3.1 as parts-of-speech (POS). For sub-sequences of POS we consider templates of length $n \in \{4, 5, 6, 7, 8\}$. Templates are characterized by their high frequency across the texts in a given corpus (e.g., one comprising texts generated by a particular LLM) Practically, we choose τ relative to the sample size. For Rotten Tomatoes we retain the top 100 most common template where the least frequent template appears 4 times in the dataset.

To extract templates we use `diversity` (Shaib et al., 2024),[2] a library providing tools to evaluate token and POS diversity in a dataset. We use this tool to first tag all tokens in a corpus with their corresponding POS tags, then search for the top 100 most frequent n-grams across these tags. `diversity` uses the SpaCy POS tagger,[3] which relies on the Penn Treebank set of 36 tags (Taylor et al., 2003). After tagging, we return frequent n-grams of POS, the corresponding matched text. Figure 2 illustrates the output of running the template extraction process.

The pipeline for identifying templates can be further extended to other tagging libraries and types. For example, we also explore constituency parses as an alternative to POS tags. Extracting templates and matching over tokens is non-trivial for constituency parses. We provide examples of the patterns identified by this abstraction in Appendix A and leave further analysis to future work.

3.3 Metrics for Measuring Templates

Our goal for extracting templates is to assess and characterize different levels of repetition in LLMs. We calculate three metrics using templates, (1) the diversity of the POS tags that are generated using CR-POS (2) the fraction of texts generated with a template using template rate and count, and (3) the number of templates that appear within each text using templates-per-token. We now elaborate on each one.

CR-POS. At the most granular level, we are interested in quantifying the n-gram diversity of the POS tag sequences present in the text. Lossless text compression algorithms—such as gZip—are optimized to detect repeated characters in sequences, and rely on this to compress documents without any loss of information. If a document contains frequent repeated strings, the document will be more compressible, resulting in a larger difference in compressed size relative to the original document size. Shaib et al. (2024) show that using gZip to calculate a compression ratio (CR) can provide an efficient measure for capturing lexical diversity, specifically n-gram repetition.

We calculate CR over a set of POS-tagged text, with higher values indicating that text is highly compressable (and therefore shows lower diversity). To calculate the CR, we concatenate all POS-tagged text into a sequence, and measure the ratio between the original document size and the compressed document size:

$$\text{CR}(f(T\oplus)) = \frac{|f(T\oplus)|}{\text{compressed } (|f(T\oplus)|)} \quad (1)$$

Where $T\oplus$ is the concatenated sequence of text, and $f(T\oplus)$ is POS-tagged text. Higher compression ratios imply more redundancy in the text, and therefore lower diversity of the sequence.

Template Rate We measure the fraction of texts in a corpus (sequence) that contain at least 1 template to quantify how frequently templates appear

[2] https://pypi.org/project/diversity/
[3] https://spacy.io/api/tagger

across an entire corpus.

$$TR = \frac{1}{T}\sum_{i=1}^{T} I_i \quad (2)$$

Where, T is the sequence of text (corpus), and

$$I_i = \begin{cases} 1 & \text{if text } i \text{ contains at least 1 template} \\ 0 & \text{otherwise} \end{cases}$$

Templates-per-Token In practice, text can contain many templates. Measures of diversity are confounded by text length (Salkar et al., 2022), which also applies to template counts; if a model tends to produce longer texts, there is a higher chance that any given output will contain a template. To compare between text sources, we can length normalize:

$$\text{TPT}(T\oplus) = \frac{\frac{1}{T}\sum_{i=1}^{T} \text{\# templates in } t_i}{\frac{1}{T}\sum_{i=1}^{T} \text{\# words in } t_i} \quad (3)$$

Where T is a concatenated sequence of tokens forming a corpus for a particular text source, and t the string.

4 Experimental Setup

4.1 Models

Open Models We first evaluate the incidence of templated text in two open-ended generation tasks using OLMo-7B Instruct (Groeneveld et al., 2024), a fully open source model that released model training checkpoints and its training data. This allows us to evaluate templates in its training datasets: Dolma (Soldaini et al., 2024), Tulu-V2 (Ivison et al., 2023), and Ultra-feedback (Cui et al., 2023).

We then evaluate templates across closed source models (which do not release training data), specifically: Mistral, Llama (-2, -3), Alpaca, and GPT-4o.

Fine-tuned (Instruction) Models We experiment with a total of 8 instruction-tuned models. We use Mistral (Instruct, 7B; Jiang et al. 2023), Alpaca (7B, 13B; Taori et al. 2023; Wang et al. 2022; Touvron et al. 2023a). In addition, 5 models are further trained on human preferences: OLMo (Instruct, 7B; Groeneveld et al. 2024), Llama-2 (Chat-HF, 7B-70B; Touvron et al. 2023b), and Llama-3 (Instruct, 70B Dubey et al. 2024).

4.2 Decoding

While greedy decoding is a common decoding strategy for many popular downstream generation tasks, one can explicitly control token diversity at inference time via choice of decoding hyperparameters such as temperature. We evaluate generation under various decoding strategies and model sizes. We refer to Wiher et al. (2022) for an in-depth discussion on the impact of sampling on generated text, and here focus specifically on varying hyperparameters and resultant impact of the appearance and frequency of templates. For the former, we use greedy decoding, and separately vary temperature and top-p for decoding with sampling. Top-p (nucleus) sample restricts the subset of tokens such that the combined probability reaches a threshold p (Holtzman et al., 2020).

4.3 Tasks and Datasets

Open-Ended Generation To evaluate intrinsic template behaviour we evaluate open-ended generation tasks in two settings. In the first setting, we sample generations from the model given only a special token denoting beginning of sequence (`[BOS]`). In the second, we randomly sample 100 tokens from Dolma and use these tokens to prompt further open-ended generation from the model.

Synthetic Data Generation LLMs are increasingly used to create synthetic training datasets, which are often used to train downstream models (e.g., Wang et al. 2022). We evaluate templates in Cosmopedia, a synthetic dataset generated by prompting Mixtral-8x7B-Instruct with instructions to produce text relating to textbooks, blogposts, stories, posts and WikiHow articles (Ben Allal et al., 2024). We prompt OLMo-7B with the Cosmopedia instructions and evaluate the resulting generations.

Summarization Summarization is a common benchmark for long text generation. We evaluate models on a handful of summarization datasets, including single- and multi-document tasks. Such datasets allow us to study templates in longer sequences that would not be evident in tasks where only a few tokens are generated. For general English-language tasks, we generate summaries and reviews over news (CNN/Daily Mail; Nallapati et al. 2016), movies (Rotten Tomatoes; Leone 2020), and books (BooookScore; Chang et al. 2023).

We also look at templates in the biomedical domain as an example of a domain-specific task. Cochrane is a dataset of systematic reviews summarizing the evidence over medical interventions

	Open Generation		Rotten Tomatoes	
Decoding Strategy	**CR: POS**	**≥ 1 Templates % ($n = 6$)**	**CR: POS**	**≥ 1 Templates % ($n = 6$)**
Greedy	702.8	100.0 (0.065)	6.45	97.0 (0.041)
Default	5.81	75.5 (0.009)	6.33	96.6 (0.041)
`temp 0.8`	6.74	71.8 (0.007)	6.26	96.6 (0.043)
`temp 0.85`	6.48	74.0 (0.007)	6.22	96.6 (0.041)
`temp 0.9`	6.22	73.4 (0.009)	6.17	97.4 (0.039)
`temp 0.95`	5.98	75.4 (0.010)	6.14	97.2 (0.040)
`top_p 0.8`	7.03	76.5 (0.007)	6.31	97.8 (0.041)
`top_p 0.85`	6.71	71.0 (0.007)	6.27	96.6 (0.041)
`top_p 0.9`	6.50	75.3 (0.008)	6.22	96.2 (0.039)
`top_p 0.95`	6.17	77.2 (0.009)	6.31	95.8 (0.041)

Table 1: Compression ratio with POS (CR-POS), average text length and percentage of generated outputs with at least 1 template of size $n = 6$, when varying `temperature` and `top_p` for OLMo-7B decoding. Arrows indicate higher template rates.

Dataset	**CR: POS**	**≥ 1 Templates % ($n = 6$)**
Dolma	5.65	82.6 (0.012)
Cosmopedia	5.76	99.1 (0.014)

Table 2: CR-POS, template-per-token, and template counts for templates of size $n = 6$ reported for OLMo-7B text generated with Cosmopedia Instructions, and 100 sampled tokens from the Dolma dataset, with greedy decoding.

(Wallace et al., 2020). We prompt models to generate systematic reviews. Importantly, these datasets include human-written reference summaries, which serve as a baseline to compare our task-specific template analysis.

5 Templates in Model-Generated Text

We first evaluate OLMo-7B Instruct on 3 tasks: open-ended generation, synthetic data generation, and summarization, using both greedy and varying temperature and top-p sampling strategies (**RQ1**).

Table 1 shows the effect of varying sampling hyperparameters temperature and top-p on the overall diversity of the generated text with OLMo-7B with open-generation and summarization. Varying sampling strategies in the open-generation task results in a higher variance of template rates (74.4% $\pm$ 2.1) compared to templates rates in the summarization task (96.8% $\pm$ 0.6). These results indicate that templatic text in summarization appears in spite of sampling strategies intended to increase (lexical) diversity. Overall, the rate of templates is much higher in the Rotten Tomatoes dataset than for open-generation, indicating downstream tasks such as summarization, which often entail prompting with instructions, may yield more repetitive structures.[4]

Table 2 shows the rate of templates on two additional tasks: Synthetic data generation and data generation with Dolma Cosmopedia results in a higher incidence of templates (99.1%) and templates per token (0.014), compared to Dolma (82.6%, 0.012).

6 Searching For Templates in Pre-training Data

One hypothesis for the emergence of templates in generated text is that these templates are overrepresented in the training data **(RQ2)**. Here we interrogate this empirically.

6.1 Emergence of Templates in Training

We first aim to understand when during training models start to generate templates. We measure the perplexity of matched texts from a set of previously extracted templates (following §3.2) across OLMo's checkpoints. Higher perplexity values indicate the templates are assigned low likelihood at that checkpoint.

For each model checkpoint, we average the perplexities of templates of length $n = 6$ and compare to the perplexities of randomly sampled 6-grams. We calculate the average perplexity for the dataset using:

$$\frac{1}{|D|}\frac{1}{|N|}\sum_{j=1}^{|D|}\sum_{k=1}^{|N|} 2^{H(p_k)} \quad (4)$$

Where N is the total number of templates in the document, and D the total number of documents in the dataset We repeat this process for randomly sampled 6-grams (distinct from the templates) to match the number of templates.

Results Figure 3 shows the average perplexities across model checkpoints. We find that templates are learned quickly—by the first model checkpoint (which was trained on 4B tokens). Average perplexity drops to around 500 for non-template tokens, compared to 200 for templates. These findings are surprising, and indicate that templates are learned early in pre-training, rather than during the fine-tuning process. The average perplexities remain lower for template tokens for the remainder of the training process.

[4]Note that we show the incidence of templates given different instructions in Appendix D

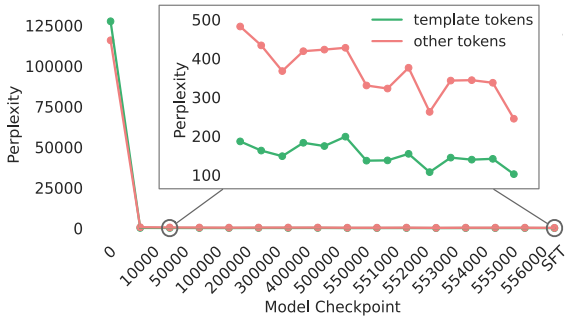

Figure 3: Perplexity of matched template at different model checkpoints. Templates initially have higher perplexity than other tokens, but quickly drop after initial training steps.

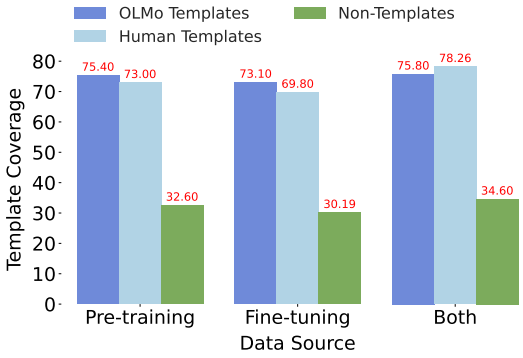

Figure 4: Coverage of templates found in pre-training data, fine-tuning data, and both datasets combined. Templates are found at a much higher rate in the training data than random n-gram sequences.

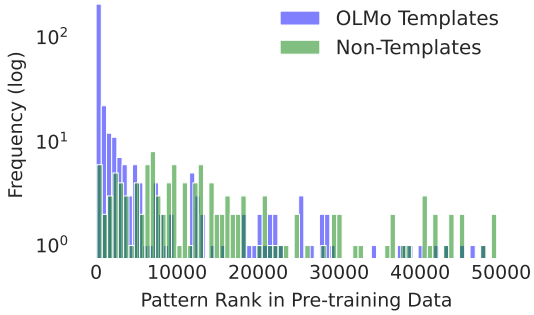

Figure 5: Ranking of OLMo templates and non-templated sequences in frequency of pre-training data. Templates appear at higher ranks (and are therefore more frequent) than non-templated sequences.

6.2 Templates in Pre-training Data

The lower perplexities in the above finding indicates that templates are seen fairly early on in pre-training compared to non-templated sequences of training data text. We next measure the incidence and types of templates in the pre-training data, and whether they correspond to the templates that models produce.

To search for template coverage by OLMo, we start by selecting a random subset of the Dolma dataset, containing 10 billion tokens. We then annotate all of the sequences with a POS tagger using the Dolma toolkit (Soldaini et al., 2024). Finally, we find the 50K most common POS-grams in the data for sequence length of six using the WIMBD toolkit (Elazar et al., 2023), which is optimized for search and count at large scales.

Results Figure 4 shows the coverage of templates produced by OLMo in the pre-training data, the fine-tuning data, and their concatenation. We find that 75% of templates produced by OLMo are found in the pre-training data, indicating that a majority of templates are not a novel construction learned during fine-tuning, Rather, they are learned directly from pre-training data. In comparison, only 34% of randomly sampled non-templated sequences are found in the pre-training data. Further, Figure 5 shows that the templates OLMo generates consistently rank higher in frequency in the pre-training dataset, compared to randomly sampled non-templates. The difference in ranks between the templates and non-templates is statistically significant; the median rank in templates and non-templates are 337.5 and 9651.0 (Mann–Whitney $U = 6043$, $p < 0.05$ two-tailed). Overall we find that, not only do most of the templates produced in downstream generation tasks appear in the pre-training data, but are also often very frequent sequences in the pre-training data.

7 Templates in Closed-Source Models

With OLMo, we find that 75% of templates are found at high frequencies in the pre-training data (§6.2). Most available models however do not release their pre-training data. Here we evaluate the incidence of templates in other closed-source models, which we define as models that do not release their training data. Addressing RQ1, we characterize the rates of templatic texts in these models, and posit that templates may be indicators of the pre-training data sources models are trained on.

Summarization We report the template rate and templates-per-token that appear in text generated by models in three summarization tasks: movie

Model	Rotten Tomatoes CR: POS	Rotten Tomatoes ≥ 1 Templates % ($n = 6$)	Cochrane CR: POS	Cochrane ≥ 1 Templates % ($n = 6$)	CNN/DM CR: POS	CNN/DM ≥ 1 Templates % ($n = 6$)
Reference	5.31	46.4 (0.040)	5.63	83.3 (0.049)	5.33	36.0 (0.013)
Input Documents	5.82	29.3 (0.001)	5.96	98.5 (0.021)	5.54	98.4 (0.020)
OLMo-7B	6.45	97.0 (0.041)	6.53	74.0 (0.030)	5.83	**91.2 (0.025)**
Mistral-7B	6.29	**99.6 (0.043)**	6.10	99.5 (0.043)	5.70	**89.9 (0.029)**
Llama-2-7B	6.87	**93.0 (0.047)**	6.43	88.4 (0.042)	5.71	**90.4 (0.028)**
Llama-2-13B	6.70	**99.0 (0.060)**	6.65	**95.1 (0.052)**	5.91	**97.4 (0.042)**
Llama-2-70B	6.36	**99.3 (0.123)**	6.51	99.7 (0.042)	5.69	**87.4 (0.027)**
Llama-3-70B	6.39	**99.2 (0.151)**	6.50	99.5 (0.030)	5.66	**83.2 (0.024)**
Alpaca-7B	6.65	**92.4 (0.070)**	7.82	**75.9 (0.051)**	6.65	**90.0 (0.027)**
Alpaca-13B	6.28	**89.2 (0.053)**	6.26	67.0 (0.043)	5.59	**85.4 (0.028)**
GPT-4o	6.11	**98.2 (0.041)**	6.12	95.7 (0.011)	5.71	**91.0 (0.026)**

Table 3: Compression ratio with POS (CR-POS) reported for each model-generated output over a random sample (n=500) of the Rotten Tomatoes, Cochrane, and CNN/DM datasets using greedy decoding, and the prompt "`Write a short summary`". For Cochrane, we use the prompt "`Write a meta-analysis`" to match the task. Larger values in CR-POS indicate *less* diversity in the sequences. We report the percentage of generated outputs with at least 1 template of size $n = 6$, and the rate of templates-per-token in parentheses (avg. num. templates per summary normalized by avg. length). Models producing higher templates-per-token than the human-written references are marked in **bold**.

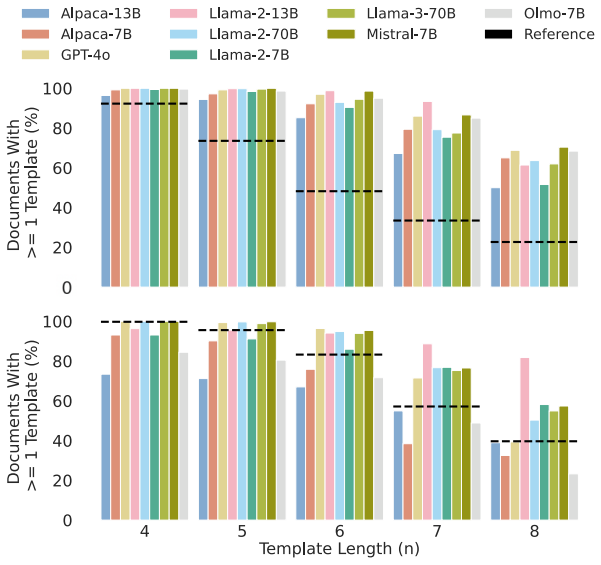

Figure 6: Incidence of generated text with at least 1 template of sizes $n = 4, 5, 6, 7, 8$ over the (a) Rotten Tomatoes and (b) Cochrane datasets. Longer templates appear less frequently but at higher rates in model-generated text than in human-written references (dashed lines).

reviews, biomedical evidence, and news (Table 3).

We find that, on average in the Rotten Tomatoes dataset, 95% of outputs contain templates of length $n = 6$ across different model types and sizes. This is in contrast to human-written reference and input documents, which contain templates of the same size on average in 38% of cases. We find a similar trend for templates of length $n = [4, 8]$ (Figure 6).

While the average number of templates is higher in model-generated output, this could be attributed to models simply producing lengthier texts than the human written references. To quantify this, we also compute the template-per-token as a length normalized value capturing the average templates per summary. Even controlling for length, most models produce more templates per token than human authors, as shown in Table 3.

The CNN/DM datasets show similar trends, but with lower rates of templates (average 89.6% contain templates) compared to the Rotten Tomatoes dataset. In contrast, the percentage of templates is high for model-generated (average 88.3%) and human-written references (83.3%) in the Cochrane dataset. This owes to the nature of meta-analysis texts, which are formulaic (Higgins and Green, 2010).[5]

Figure 6 illustrates the rate of templates for each model as the template length grows from length $n = 4$ to 8 for Rotten Tomatoes and Cochrane. For Rotten Tomatoes (and CNN; Appendix B), all models produce templates at higher rates than human-written summaries across all template lengths. With Cochrane, template lengths ≥ 6 show the majority of models produce higher rates of templates than human authored references. This indicates that differences between templatedness in human-authored references and LLM summaries surface only at longer template lengths.

[5]Table C in the appendix provides examples of the human-written references

Model	BooookScore, Hierarchical CR: POS	≥ 1 Templates % ($n = 6$)
Claude-2048	5.63	95.0 (0.010)
Claude-88000	5.60	94.0 (0.004)
ChatGPT-2048	6.17	100.0 (0.017)
GPT4-2048	6.04	100.0 (0.013)
GPT4-4096	6.01	99.0 (0.013)
Mixtral-2048	6.01	100.0 (0.017)

Table 4: Compression ratio with POS (CR-POS) reported for the BooookScore dataset. We report the percentage of generated outputs with at least 1 template of size $n = 6$, and the rate of templates-per-token in parentheses.

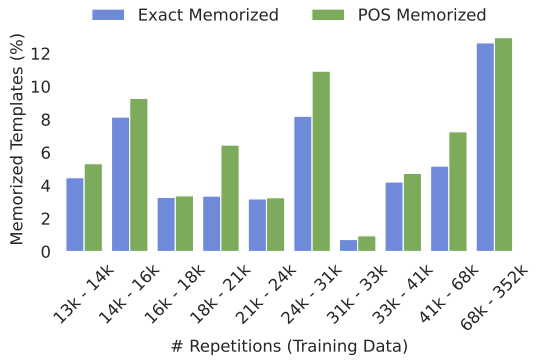

Figure 7: Percent memorized POS sequences, stratified by frequency in training data.

The BooookScore dataset provides text generated by models using long-document summarization strategies. We report results over the hierarchical strategy, where the final summary is merged together from smaller summarized chunks. Similar to the other summarization tasks, we observe high rates of templates across all the models available in this dataset (Table 4).

Effect of Model Size on Template Rates Table 3 reports differences in the rate of templates between different sizes of Llama-2 and Alpaca. In the case of Alpaca, the larger model yields outputs with less repetition and fewer templates. With the Llama-2 and Llama-3 models, we observe a surprising trend as model size increases: CR-POS and average text length decrease, however the rate of summaries that contain one or more templates stays the same (and increases slightly in some cases). These results indicate that larger models do not necessarily produce less templated outputs. The templates-per-token value further supports this, showing an increase in template rate (per token) for larger models.

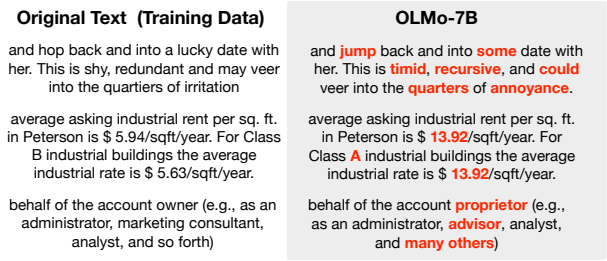

Figure 8: Sampled sentences from OLMo-7B prompted with a prefix of training data. The model substitutes synonyms or numbers.

8 Style Memorization

Past work has shown that models memorize portions of pre-training data. Carlini et al. (2022) show a lower bound of 1% verbatim memorized data. We next show how our syntactic template analysis can be used to evaluate how much models memorize from pre-training data, beyond strict token sequence matches (**RQ3**).

Definition: Exact-Text Memorization We borrow the definition for extractable memorization from Carlini et al. (2022). A string s is memorized by a model $\mathcal{M}$ if, when prompted with context p, $\mathcal{M}(p)$ produces a string g that is an exact text match to the source string s under greedy decoding.

Definition: Style (POS) Memorization For style memorization, we follow the same definition for Exact-Text, but modified to operate over POS (rather than token) sequences to capture instances of syntactic "style". Specifically, given a POS tagger f, sequence $f(s)$ is memorized by $\Updownarrow$ if, when prompted with context p, $\Updownarrow(p)$ produces a sequence g such that $f(g)$ is an exact match to the source string $f(s)$.

8.1 Experimental Setup

We follow a similar experimental setup as Carlini et al. (2022), focusing on creating sampled datasets that contain n-grams repeated in the pre-training dataset. We use WIMBD to build our subset over the Dolma dataset and return the top 50k most common 100-grams in the Dolma dataset from § 6.2. For each 100-gram, we tokenize the sequence with NLTK and truncate to 50 tokens (Carlini et al., 2022). Following the setup for extracting memorized sequences, we prompt OLMo-7B with 50 tokens of the training data sequence and generate a maximum of 1000 tokens using greedy decoding. We apply NLTK's POS model to tag the original

string s and the model-generated string g.

8.2 Results

We randomly sample 10k documents and look at the fraction of memorized outputs based on exact-match and the POS sequence ("style") memorization using the `diversity` package (e.g., Fig. 2). We average the fraction memorized over 1,000 seeds for sampling the datasets.

On average, the POS memorization definition finds 6.4% ($\pm$ 0.7) memorized, whereas exact text match only reports 5.3% ($\pm$ 0.6) memorized of the training dataset. Figure 7 shows the percent templates memorized stratified by frequency of the 100-gram in the training dataset. We divide the sampled data point into 10 buckets each containing 4,138 samples with counts that fall in each bucket. In all buckets, POS memorization captures a higher rate of memorized sequences.

The implications of our looser definition of memorization allows us to capture instances of memorization where exact tokens may be substituted during generation, but where an output span is nonetheless structurally the same as a source string. Note that this method will by default also capture duplicate text in addition to softly memorized sequence. In Figure 8, we provide sampled examples of substitutions that occur that are *not* captured by exact-memorization definitions, yet demonstrate that the particular style of that training point has been memorized. We find that these cases often include synonym swaps, or different numbers being generated.

9 Conclusions

In this work, we introduce syntactic templates as a framework for analyzing subtle repetitive characteristics in model-generated text. We show that this analysis can also extend to human-written references and downstream tasks, and find that the pre-training data contains many of these identified templates. We show that evaluating repetition in parts-of-speech sequences is useful for detecting subtle types of data "memorization". Our hope is that this work inspires additional research into characterizing where (in data) observed stylistic patterns in LLM outputs originate.

Limitations

There are a few limitations to this work that we address here.

First, this type of analysis requires an entire corpus that is representative of a text source. For paid models, this can be costly to obtain. For large datasets, this can be resource intensive. These considerations provide a potential barrier based on available resources.

Second, we use third party tools to tag our text abstractions; however these tools are deterministic, but can contain errors in the tags they assign to sequences, particularly if a sequence contains text from another language. We assume that the majority of the text we analyze is in English, and that any errors are superseded by the frequency of common templates.

Finally, this work only examines English texts, in part due to availabilty of datasets at the scale necessary to evaluate models.

Acknowledgements

We thank Niloofar Mireshghallah for guidance on the memorization experiments, Kyle Lo, and Luca Soldaini for their advice and assistance on accessing OLMo and pretraining data. This work was supported in part by the National Science Foundation (NSF) grants IIS 2211954, IIS 2145479, IIS 2107524.

References

Kevin Bache, David Newman, and Padhraic Smyth. 2013. Text-based measures of document diversity. *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*.

Daniel Bär, Torsten Zesch, and Iryna Gurevych. 2012. Text reuse detection using a composition of text similarity measures. In *Proceedings of COLING 2012*, pages 167–184.

John Bauer, Chloé Kiddon, Eric Yeh, Alex Shan, and Christopher D. Manning. 2023. Semgrex and ssurgeon, searching and manipulating dependency graphs. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories (TLT, GURT/SyntaxFest 2023)*, pages 67–73, Washington, D.C. Association for Computational Linguistics.

Loubna Ben Allal, Anton Lozhkov, Guilherme Penedo, Thomas Wolf, and Leandro von Werra. 2024. Cosmopedia.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

Yapei Chang, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2023. Booookscore: A systematic exploration of book-length summarization in the era of llms. *ArXiv*, abs/2310.00785.

Mayee Chen, Nicholas Roberts, Kush Bhatia, Jue Wang, Ce Zhang, Frederic Sala, and Christopher Ré. 2024. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Chrysanne DiMarco and Graeme Hirst. 1993. A computational theory of goal-directed style in syntax. *Computational Linguistics*, 19(3):451–500.

Yao Dou, Maxwell Forbes, Rik Koncel-Kedziorski, Noah A Smith, and Yejin Choi. 2022. Is gpt-3 text indistinguishable from human text? scarecrow: A framework for scrutinizing machine text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood,

Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models.

Yanai Elazar, Akshita Bhagia, Ian Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, et al. 2023. What's in my big data? *arXiv preprint arXiv:2310.20707*.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, A. Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Daniel Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hanna Hajishirzi. 2024. Olmo: Accelerating the science of language models. *ArXiv*, abs/2402.00838.

Julian P T Higgins and Sally Green. 2010. Cochrane handbook for systematic reviews of interventions. *International Coaching Psychology Review*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Dieuwke Hupkes, Mario Giulianelli, Verna Dankers, Mikel Artetxe, Yanai Elazar, Tiago Pimentel, Christos Christodoulopoulos, Karim Lasri, Naomi Saphra, Arabella Sinclair, Dennis Ulmer, Florian Schottmann, Khuyagbaatar Batsuren, Kaiser Sun, Koustuv Sinha, Leila Khalatbari, Maria Ryskina, Rita Frieske, Ryan Cotterell, and Zhijing Jin. 2023. A taxonomy and review of generalization research in nlp. *Nature Machine Intelligence*, 5(10):1161–1174.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew E. Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hanna Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *ArXiv*, abs/2311.10702.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier,

L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *ArXiv*, abs/2310.06825.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2024. Understanding the effects of rlhf on llm generalisation and diversity.

Kalpesh Krishna, John Wieting, and Mohit Iyyer. 2020. Reformulating unsupervised style transfer as paraphrase generation. *arXiv preprint arXiv:2010.05700*.

Wojciech Kryściński, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Stefano Leone. 2020. Rotten tomatoes movies and critic reviews dataset.

Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, et al. 2024a. Monitoring ai-modified content at scale: A case study on the impact of chatgpt on ai conference peer reviews. *arXiv preprint arXiv:2403.07183*.

Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuandong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, et al. 2024b. Mapping the increasing use of llms in scientific papers. *arXiv preprint arXiv:2404.01268*.

R. Thomas McCoy, Paul Smolensky, Tal Linzen, Jianfeng Gao, and Asli Celikyilmaz. 2023. How Much Do Language Models Copy From Their Training Data? Evaluating Linguistic Novelty in Text Generation Using RAVEN. *Transactions of the Association for Computational Linguistics*, 11:652–670.

William Merrill, Noah A. Smith, and Yanai Elazar. 2024. Evaluating n-gram novelty of language models using rusty-dawg.

Ehsan Montahaei, Danial Alihosseini, and Mahdieh Soleymani Baghshah. 2019. Jointly measuring diversity and quality in text generation models. *ArXiv*, abs/1904.03971.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Vishakh Padmakumar and He He. 2023. Does writing with language models reduce content diversity? *arXiv preprint arXiv:2309.05196*.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Nicholas Roberts, Davis Liang, Graham Neubig, and Zachary C Lipton. 2020. Decoding and diversity in machine translation. *arXiv preprint arXiv:2011.13477*.

Ariel Rosenfeld and Teddy Lazebnik. 2024. Whose llm is it anyway? linguistic comparison and llm attribution for gpt-3.5, gpt-4 and bard. *arXiv preprint arXiv:2402.14533*.

Nikita Salkar, Thomas Trikalinos, Byron Wallace, and Ani Nenkova. 2022. Self-repetition in abstractive neural summarizers. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 341–350, Online only. Association for Computational Linguistics.

Chantal Shaib, Joe Barrow, Jiuding Sun, Alexa F Siu, Byron C Wallace, and Ani Nenkova. 2024. Standardizing the measurement of text diversity: A tool and a comparative analysis of scores. *arXiv preprint arXiv:2403.00553*.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, et al. 2024. Dolma: An open corpus of three trillion tokens for language model pretraining research. *arXiv preprint arXiv:2402.00159*.

Juan Soler-Company and Leo Wanner. 2017. On the relevance of syntactic and discourse features for author profiling and identification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 681–687, Valencia, Spain. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The penn treebank: an overview. *Treebanks: Building and using parsed corpora*, pages 5–22.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open

and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Byron C. Wallace, Sayantani Saha, Frank Soboczenski, and Iain James Marshall. 2020. Generating (factual?) narrative summaries of rcts: Experiments with neural multi-document summarization. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2021:605–614.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*.

Gian Wiher, Clara Meister, and Ryan Cotterell. 2022. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F Wong, and Lidia S Chao. 2023. A survey on llm-gernerated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Template (Constituents)	Frequency	Matched Text
(VP (VB) (NP (NP (DT) (JJ) (NN)) (PP (IN) (NP)	458	Trace the intellectual history of ancient Examine the early history of automobiles guarantee a seamless ascent into another reach a broad audience of buyers
(PP (IN) (NP (NP (DT) (JJ) (NN)) (PP (IN) (NP)	948	as a key component of your by a high abundance of free across a global range of cultures for the comprehensive study of the
(DT) (JJ) (NN)) (PP (IN) (NP)	1680	a strong grasp of various a solid understanding of a radical change in

Table 5: Example templates using constituency trees over the Cosmopedia dataset.

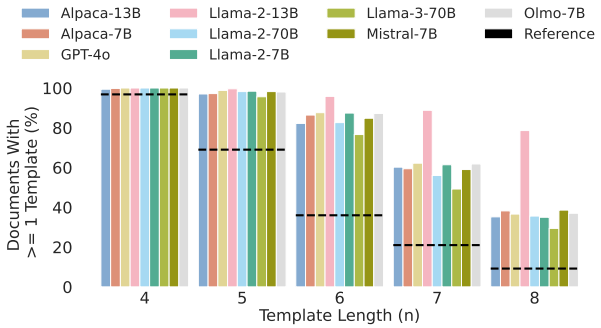

Figure 9: Incidence of generated text with at least 1 template of sizes $n = 4, 5, 6, 7, 8$ over the CNN/DM dataset.

A Constituency Trees

For constituency parsing, we use the Stanza library (Bauer et al., 2023), and linearize sequences using a breadth-first search approach. Appendix A shows some examples of linearized constituency trees and their matching text.

B CNN/DM Trends

Figure 9 demonstrates the same trend of high templatic text in the model generated text compared to human authored references.

Prompts, CNN/DM	
1	Please write a summary.
2	Please write a summary of the article.
3	Summarize.
4	Write a short summary.
5	Write a short summary and be creative.
6	Write a meta-analysis.
7	Write an aggregate summary based on the above facts.

Table 6: Prompts used for the CNN/DM summarization task.

Prompts, Cochrane	
1	Please write a summary.
2	Please write a summary of the evidence.
3	Summarize.
4	Write a short summary.
5	Write a short summary and be creative.
6	Write a meta-analysis.
7	Write an aggregate analysis based on the above evidence.

Table 7: Prompts used for the Cochrane meta-analysis task.

There is no evidence from good quality randomized trials or non-randomized studies of the effectiveness of lens extraction for chronic primary angle-closure glaucoma.
There was not enough evidence to judge whether or not the included drugs cured bedwetting when used alone. [...] There was also evidence to suggest that combination therapy with anticholinergic therapy increased the efficacy of other established therapies such as imipramine, desmopressin and enuresis alarms by [...]. Future studies should evaluate the role of combination therapy against established treatments in rigorous and adequately powered trials.
There is some evidence for use of botulinum toxin injections to salivary glands for the treatment of sialorrhea in MND. Further research is required on this important symptom. Data are needed on the problem of sialorrhea in MND and its measurement, both by patient self report measures and objective tests. These will allow the development of better randomized controlled trials.
There is significant evidence to suggest that topical application of chlorhexidine to umbilical cord reduces neonatal mortality and omphalitis in community and primary care settings in developing countries. It may increase cord separation time however, there is no evidence that it increases risk of subsequent morbidity or infection. There is insufficient evidence to support the application of an antiseptic to umbilical cord in hospital settings compared with dry cord care in developed countries.
We found insufficient evidence to determine if overground physical therapy gait training benefits gait function in patients with chronic stroke, though limited evidence suggests small benefits for variables such as gait speed or 6MWT. These findings must be replicated by large, high quality studies using varied outcome measures.

Figure 10: Sample of Cochrane (Human-Authored) references.

C Cochrane References

Cochrane systematic reviews follow guidelines for how they should be written. Figure 10 shows an example of human-authored Cochrane text.

D Choice of Prompt

The choice of prompt can impact the incidence of templates. Here, we provide an analysis of 7 different prompts for the downstream summarization task. We find that shorter and more generic prompts

Prompts, Rotten Tomatoes	
1	Please write a summary.
2	Please write a summary of the movie reviews.
3	Summarize.
4	Write a short summary.
5	Write a short summary and be creative.
6	Write a meta-analysis.
7	Write an aggregate movie review based on the above reviews.

Table 8: Prompts used for the Rotten Tomatoes summarization task.

Model	≥ 1 Templates % ($n = 6$)							
Prompt	**1**	**2**	**3**	**4**	**5**	**6**	**7**	Δ
OLMo-7B	98.1	94.4	92.5	97.0	96.2	97.2	97.2	5.6
Mistral-7B	98.5	98.5	97	99.6	99.5	100	100	3.0
Llama-2-7B	91.5	94.5	88.9	93.0	94.8	95.4	94.9	6.5
Llama-2-13B	99.1	99.8	94.9	99.0	98.0	98.0	98.6	4.9
Llama-2-70B	96.2	96.8	97.2	99.3	98.8	99.6	99.2	3.4
Llama-3-70B	95.7	97.3	93.7	99.2	97.8	99.8	99.4	6.1
Alpaca-7B	92.3	87.5	86.8	92.4	92.7	84.1	91.6	8.6
Alpaca-13B	92.5	91.0	90.3	89.2	91.6	94.5	94.4	5.3
GPT-4o	98.2	98.0	95.2	97.0	99.6	100.0	99.8	4.8

Table 9: Percentage of generated summaries with at least 1 template of length $n = 6$ under different instruction prompts. Some instruction prompts result is more templates outputs for certain models.

result in lower performance. It is possible that with the shorter and more generic instructions instructions, there may be a higher overlap in template types across models.

The choice of prompt can affect the rate of templates. Appendix Table 9 shows the rate of templates with different prompts while the choice of prompt impacts the rate of templates, the incidence remains on average higher than 90%.

E Template Measures, Text Length

For completeness, we provide the full tables for all datasets that include the average text length. In the case of Booookscore, we also provide results over the incremental dataset.

Model	Rotten Tomatoes			Cochrane			CNN/DM		
	CR: POS	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**	**CR: POS**	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**	**CR: POS**	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**
Reference	5.31	25.1	46.4 (0.040)	5.63	73.8	83.3 (0.049)	5.33	57.7	36.0 (0.013)
Input Documents	5.82	668.2	29.3 (0.001)	5.96	1555.3	98.5 (0.021)	5.54	514.7	98.4 (0.020)
OLMo-7B	6.45	194.4	97.0 (0.041)	6.53	158.1	74.0 (0.030)	5.83	177.1	**91.2 (0.025)**
Mistral-7B	6.29	185.6	**99.6 (0.043)**	6.10	177.2	99.5 (0.043)	5.70	153.0	**89.9 (0.029)**
Llama-2-7B	6.87	126.5	**93.0 (0.047)**	6.43	151.0	88.4 (0.042)	5.71	153.9	**90.4 (0.028)**
Llama-2-13B	6.70	117.0	**99.0 (0.060)**	6.65	157.7	**95.1 (0.052)**	5.91	143.5	**97.4 (0.042)**
Llama-2-70B	6.36	114.2	**99.3 (0.123)**	6.51	324.7	99.7 (0.042)	5.69	138.3	**87.4 (0.027)**
Llama-3-70B	6.39	106.1	**99.2 (0.151)**	6.50	387.5	99.5 (0.030)	5.66	132.7	**83.2 (0.024)**
Alpaca-7B	6.65	99.4	**92.4 (0.070)**	7.82	98.1	**75.9 (0.051)**	6.65	145.2	**90.0 (0.027)**
Alpaca-13B	6.28	93.0	**89.2 (0.053)**	6.26	69.7	67.0 (0.043)	5.59	138.1	**85.4 (0.028)**
GPT-4o	6.11	203.5	**98.2 (0.041)**	6.12	560.7	95.7 (0.011)	5.71	167.6	**91.0 (0.026)**

Table 10: Compression ratio with POS (CR-POS) reported for each model-generated output over a random sample (n=500) of the Rotten Tomatoes, Cochrane, and CNN/DM datasets using greedy decoding, and the prompt "`Write a short summary`". For Cochrane, we use the prompt "`Write a meta-analysis`" to match the task. Larger values in CR-POS indicate *less* diversity in the sequences. We report the percentage of generated outputs with at least 1 template of size $n = 6$, and the rate of templates-per-token in parentheses (avg. num. templates per summary normalized by avg. length). Models producing higher templates-per-token than the human-written references are marked in bold.

Decoding Strategy	Open Generation			Rotten Tomatoes		
	CR: POS	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**	**CR: POS**	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**
Greedy	702.8	48.3	100.0 (0.065)	6.45	194.4	97.0 (0.041)
Default Sampling	5.81	479.2	75.5 (0.009)	6.33	194.2	96.6 (0.041)
`temp 0.8`	6.74	517.9	71.8 (0.007)	6.26	191.1	96.6 (0.043)
`temp 0.85`	6.48	506.8	74.0 (0.007)	6.22	188.5	96.6 (0.041)
`temp 0.9`	6.22	497.3	73.4 (0.009)	6.17	190.3	97.4 (0.039)
`temp 0.95`	5.98	500.2	75.4 (0.01)	6.14	186.6	97.2 (0.040)
`top_p 0.8`	7.03	541.0	76.5 (0.007)	6.31	190.2	97.8 (0.041)
`top_p 0.85`	6.71	526.3	71.0 (0.007)	6.27	190.3	96.6 (0.041)
`top_p 0.9`	6.50	495.1	75.3 (0.008)	6.22	190.9	96.2 (0.039)
`top_p 0.95`	6.17	513.0	77.2 (0.009)	6.31	192.1	95.8 (0.041)

Table 11: Compression ratio with POS (CR-POS), average text length and percentage of generated outputs with at least 1 template of size $n = 6$, when varying `temperature` and `top_p` for OLMo-7B decoding.

Model	BooookScore, Hierarchical			BooookScore, Incremental		
	CR: POS	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**	**CR: POS**	**Avg. Text Length**	**≥ 1 Templates % ($n = 6$)**
Claude-2048	5.63	461.9	95.0 (0.010)	5.71	575.4	86.0 (0.007)
Claude-88000	5.60	487.9	94.0 (0.004)	5.60	439.5	96.0 (0.010)
ChatGPT-2048	6.17	613.8	100.0 (0.017)	5.76	439.2	95.0 (0.011)
GPT4-2048	6.04	706.5	100.0 (0.013)	5.95	721.0	99.0 (0.009)
GPT4-4096	6.01	855.4	99.0 (0.013)	6.05	1,013.2	100.0 (0.010)
Mixtral-2048	6.01	619.2	100.0 (0.017)	5.59	496.1	99.0 (0.012)

Table 12: Compression ratio with POS (CR-POS) reported for the BooookScore dataset. We report the percentage of generated outputs with at least 1 template of size $n = 6$, and the rate of templates-per-token in parentheses.

Dataset	CR: POS	Avg. Text Length	≥ 1 Templates % ($n = 6$)
Dolma-100	5.65	483.2	82.6 (0.012)
Cosmopedia	5.76	768.0	99.1 (0.014)

Table 13: CR-POS, template-per-token, and template counts for templates of size $n = 6$ reported for OLMo-7B text generated with Cosmopedia Instructions, and 100 sampled tokens from the Dolma dataset, with greedy decoding.